Performance Optimization of Reconfigurable Real-Time Wireless Sensor Networks

Maroua Gasmi, Olfa Mosbahi, Mohamed Khalgui, Luis Gomes and Zhiwu Li, *Fellow*, *IEEE*

***Abstract*—Wireless sensor networks can be seriously impacted by several changes in behavior. As a measure of optimality, a network should react in real-time. However, even if some reactions can make the system flexible, they can cause significant damages when they are not well-structured. Moreover, the scenarios of reconfigurations can affect several parameters within a wireless sensor network. Among these parameters it is possible to name the energy efficiency, the memory limitation within a node, the real-time constraints at the level of the nodes and the network in general. Either through the application or the transmission of the reconfiguration scenarios, some parameters are severely altered. A metamorphosis of the internal architecture of the nodes is proposed in this research work as well as a policy of communication between several nodes aiming to adapt the network to any change. The resulting proposition offers general efficiency in energy and real-time constraints. The efficiency is realized by the application of a pipelined approach, dealing with the incoming reconfiguration scenarios, and a communication protocol, based on a Priority Energy and Deadline Aware Scheduling Algorithm (a multi-criteria scheduling algorithm). This protocol is capable of optimizing the transmission of both reconfiguration scenarios and sensed data simultaneously. The solution is tested on a network connecting several cars during an automobile journey and the results validate the efficiency of this proposition.**

***Index Terms*—Wireless sensor network, Reconfiguration, Energy, Communication protocol, Middleware.**

I. INTRODUCTION

The combination of sensing, computing and communicating information within a tiny device is the essential functionality of wireless sensor networks. Due to the several advances in the networking protocols, these devices form a sea of connectivity that extends the reach of cyberspace out into the physical world [1].

The advances in this network, as a part of the new generation of the embedded systems, impose real-time constraints on its functioning procedures [4]. As any real-time system, the performed services are executed by a specified number of tasks. Real-time embedded tasks are generally executed under **(i)** time constraints represented by deadlines, **(ii)** limited memory [5], and **(iii)** batteries that provide low energy [9].

Because of the variation in their applications and the increase in their requirements, the design of wireless sensor networks keeps evolving. The concept of reconfigurations is considered as the recent and more effective evolution [10]. The introduced concept of reconfigurations is related to the mode change theory [11]. In this theory, a system is offered by the possibility of switching between modes of execution. Within a specific mode, removing/adding tasks and/or changing their parameters are the basic procedures that take place [37]–[42], [47]. However, the sudden changes required by the reconfiguration scenarios can severely alter the functions of the nodes.

In dynamic environments, the reconfiguration and self-adaptation are vital capabilities of sensor networks established by varying their functional and extra-functional (real-time and energy) properties [44], [46]. Therefore, software architectural solutions are usually used to monitor and guide the dynamic changes to any WSN oriented application [43], [45], [48].

DISON reported in [32] adopts a policy-based reasoning (PBR) approach to provide a generic management system for wireless snesor networks. Its main goal is to allow sensor nodes to adapt autonomously to changes in application requirements and network resources. A multilevel management mechanism is used where every sensor node is empowered to participate in the management process at different levels according to their resources. A management function in DISON aims to monitor network resources, detecting faults, and reconfiguring nodes operation.

The authors in [33] describe a QoS-driven self-adaptive architecture for a WSN. This architecture, based on feedback control loop (FCL), adopts Monitor-Analyze-Plan-Execute over a shared knowledge (MAPE-K). Generally, WSN middleware systems that apply this approach use hierarchical networks.

Starfish described in [34] is a pattern-based representation (PBR) system proposed for specifying and dynamically managing policies in sensor nodes. They include a policy system

[0]This research paper stems from the collaboration between LISI Lab (INSAT at University of Carthage in Tunisia), PASRI (Ministry of High Study and Research in Tunisia), Universidade Nova de Lisboa in Portugal and Complex Systems Lab (Jinan University in China). It is financially supported as a MOBIDOC grant from the European Commission and also supported by the Science and Technology Development Fund, MSAR, under Grant No. 078/2015/A3 (*Co-corresponding authors: M. Khalgui and Z. Li*).

[1]M. Gasmi is with School of Electrical and Information Engineering, Jinan University, China, with the Faculty of Mathematics, Physical and Natural Sciences (FST), University Tunis-El Manar, Tunis 2092, Tunisia, and also with the National Institute of Applied Sciences and Technology (INSAT), University of Carthage, Tunis 1080, Tunisia (e-mail: mra.gsm@gmail.com).

[2]O. Mosbahi is with the National Institute of Applied Sciences and Technology (INSAT), University of Carthage, Tunis 1080, Tunisia (e-mail: olfamosbahi@gmail.com).

[3]M. Khalgui is with is with School of Electrical and Information Engineering, Jinan University, China, and also with the National Institute of Applied Sciences and Technology (INSAT), University of Carthage, Tunis 1080, Tunisia (e-mail: khalgui.mohamed@gmail.com).

[4]L. Gomes is with Universidade Nova de Lisboa-FCT-DEE and UNINOVA-CTS, P-2829-516, Monte de Caparica, Portugal (e-mail: lugo@fct.unl.pt).

[5]Z. Li is with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa 999078, Macau, China, and also with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China (e-mail: zhwli@xidian.edu.cn).

to specify dynamic adaptations.

The work reported in [19] proposes a real time wireless media access control protocol based on earliest deadline first scheduling scheme. Unlike its predecessors, this new protocol is applicable to both event-driven and clock-driven nodes with salient features such as high energy efficiency and the priority based latency.

The authors in [20] show a new approach for network scheduling in home automation applications based on IEEE 802.15.4 wireless sensor networks. This work addresses several advantages due to the introduction of rate monotonic policy for guaranteed time slots (GTSs) allocation combined with priority-based Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) for latencies reduction on transmission attempts.

The work in [35] introduces the embedded virtual machine (EVM), a programming abstraction where controller tasks with their control and timing properties are maintained across physical node boundaries and the functionality is capable of migrating the most competent set of physical controllers. In the context of process and discrete control, an EVM is a distributed run-time system that dynamically selects primary-backup sets of controllers given spatial and temporal constraints of the underlying wireless network. EVM-based algorithms allow network control algorithms to operate seamlessly over less reliable wireless networks with topological changes. They introduce new capabilities such as predictable outcomes during sensor/actuator failures, adaptation to mode changes, and run-time optimization of the resource consumption.

Agilla reported in [36] is a mobile agent middleware designed to support self-adaptive applications in wireless sensor networks. It provides a programming model in which applications consist of evolving communities of agents that share a wireless sensor network. Coordination among the agents and access to physical resources are supported by a tuple space abstraction. Agents can dynamically enter and exit a network and can autonomously clone and migrate themselves in response to environmental changes. Agilla's ability to support self-adaptive applications in wireless sensor networks has been demonstrated in the context of several applications, including fire detection and tracking, monitoring cargo containers, and robot navigation.

Contrariwise, the approach proposed in this paper treats the application and transmission of reconfigurations within a wireless sensor network differently. A node is self-sufficient in a way that it manages several incoming reconfiguration requests without wasting too much power or exhausting the internal memory. As for the medium of transmission, it is based on reloading the energy at predictable moments predefined in an offline mode. Thus a general approach is brought to the world that is energy efficient and real-time respecting as well.

At run-time, the successive requests are able of destabilizing the system and causing a general confusion. In addition, this phenomenon can have an impact on the memory by overflowing it and the energy by consuming more and more power. When the deadlines of some tasks are violated because of the application of a reconfiguration scenario at run-time, it is possible to say that the real-time aspect of the node is a victim.

Similarly, the conservation in power consumption is considered as the main challenge that a wireless sensor network can face. In fact, being spread apart in distant locations, the nodes need to be monitored frequently to insure their sufficiency in terms of energy. Moreover, the transmission procedure can be the principal responsible for similar leaks. Consequently, the low-power study within the wireless sensor network takes a major interest.

Overloading the memory of a node is a phenomenon that can also crop up as a result to the application of several reconfiguration scenarios. However, when trying to solve all of the previously mentioned problems, the preferences of the administrator can be simply ignored.

Several studies are proposed to solve the mentioned problems. The works in [32], [34] take interest in dealing with the adaptation of the wireless sensor networks at run-time. While other works in [33] consider the power management within this network. However, none of them deals with finding a general solution for the combination of the previously mentioned problems. Unlike these existing works, the proposed approach offers a generic resolution of the problems not only at the level of a single node but also for the network in general. Previous works such as the pipeline of reconfigurations [16] and the scheduling algorithm PEDASA [17] are implemented in the actual approach in order to solve the mentioned issues. This current proposition is a global solution to the problems occurring during transmission of reconfiguration scenarios simultaneously with the sensed data. Therefore, the pipeline of reconfigurations is introduced as a module in the proposed middleware, interacting with the other suggested modules. As for the PEDASA algorithm, it has been offering a solution to the partition of data within the superframes.

The pipeline of reconfigurations plays the role of a middleware within the node and guarantees the respect of the memory and power constraints during the application of several reconfiguration scenarios. This pipeline is a set of consecutive steps, each of which controls features related to the incoming scenarios of reconfiguration. At the network level, in order to synchronize the transmission of the several types of data, a new communication protocol is also suggested. This protocol based on the PEDASA approach considers the medium as a real-time processor and the sent messages as tasks having as major parameters deadline, priority and amount of consumed energy. A frame is constructed based on the decision calculation that finds a compromise between the three parameters in order to organize the set of packets in a specific order. To allow these propositions to take place, a hardware implementation is also taken into consideration offering a supply circuit allowing the dynamic reloading of the node using a solar panel technique.

This approach, as a part of the European PhD project known as MOBIDOC [1], is applied to a WSN used for the automobile journey conducted as an experiment by the embedded department in the company ARDIA [2]. The added values of this study

[1] www.anpr.tn/mobidoc/

[2] www.ardia.com.tn

are pinpointed through this implementation, in addition to its comparison with the works reported in [19], [20], [35], [36]. In fact, even the research works reported in [35], [36] support the reconfiguration feature, the considered nodes are self-adaptive in a way that they can be overloaded by extra efforts. However, this case is externalized in the proposed approach that allows to control the network changes. Moreover, the energy conservation is considered in [19], [36] by managing the sleeping modes in the nodes. Unlike these propositions, the given approach considers extending the lifespan of the node by offering an original technique that allows recharging the energy at a well specified time. Furthermore, except for Agilla [36], all the studied approaches consider real-time techniques during the data transmission. On the other hand, the approach proposed in this paper considers PEDASA as a real-time protocol during transmission. This protocol proved its efficiency compared with related techniques as demonstrated in a previous work [17]. To resume, the current paper's originality lies in:

- Controlling the memory allocation during the application of the reconfiguration scenarios,
- Expanding the lifespan of the network, by conserving the energy during computations and recharging the power wisely,
- Respecting the real-time constraints during both transmission and computation and,
- Allowing an external administrator to monitor the adaptation of the network.

In order to expose this approach effectively, this paper is organized as follows: the following section presents the basic assumptions that formalize the nodes and the mediums. Section 3 models the problems into two parts : node related problems and medium related problems. Moreover the proposed approach is defined in section 4. Therefore, the software, network and hardware architectures are exposed in this part. Finally a case study is conducted to present the actual problems that are solved through the implementation of this solution.

II. Basic Assumptions

This section describes the mathematical representations of each mentioned element in the system. A wireless sensor network is characterized by a set of *N* linear nodes connected via a medium. Let $Node_x$ and $Node_y$ be two adjacent nodes in the network. The medium that connects these two nodes is denoted by Med_{xy}. Consequently, the assumptions made in this paper are divided into two groups: the parameters of nodes and the medium characterizations.

A. Node Formalization

A sensor $Node_x$ ($x \in [1..N]$) is a real-time system. We use Ω_x and R_x to denote respectively the sets of software tasks that can be executed and the hardware/software resources existing in the node. Consequently, the description of the global environment that a node represents is given by Equation (1).

$$Node_x = (\Omega_x, R_x) \tag{1}$$

As previously mentioned, the reconfiguration is the scenario of adding (activating) or removing (deactivating) tasks in the system. Therefore, Equation (2) describes the actions related to the addition and removal of tasks.

$$\Omega_x(t) = \begin{cases} \Omega_x(t^-) \cup \xi_x(t) \text{ if addition} \\ \\ \Omega_x(t^-) \setminus \Delta_x(t) \text{ if removal} \end{cases} \tag{2}$$

where $\xi_x(t)$ and $\Delta_x(t)$ are respectively the sets of tasks to be added (activated) and removed (deactivated) from $Node_x$. Let *Reconf* be the reconfiguration scenario responsible for the previously mentioned action. Moreover, the parameters of this request are mainly: **(i)** η informs if the reconfiguration scenario adds or removes tasks; **(ii)** θ represents the priority of the reconfiguration. Two reconfigurations where one adds a certain task while the other removes it are dependent; **(iii)** β represents the list of dependent reconfiguration scenarios; **(iv)** Γ added or removed tasks. This last set can contain either $\xi_x(t)$ or $\Delta_x(t)$.

The level of the remaining energy in a node is divided into $E_{T_x}(t)$, $E_{R_x}(t)$ and $E_{C_x}(t)$. These parameters refer respectively to the level of remaining power at a moment *t* dedicated to the transmission, reception and internal computations. The node is also characterized by the availability of the memory to be allocated by the real-time tasks. The addition of tasks decreases the size of the memory while their removal increases it. Let $M_x(t)$ be the amount of available memory in $Node_x$ at the moment *t* and $M_{\tau_i^x}$ be the memory consumed by a specific task τ_i^x. Equation (3) describes the change in the previously mentioned level according to the addition or the removal of tasks in the system after a reconfiguration scenario.

$$M_x(t) = \begin{cases} M_x(t^-) - \sum_{i=1}^{n_x} M_{\tau_i^x} \text{ if tasks are added} \\ \\ M_x(t^-) + \sum_{i=1}^{n_x} M_{\tau_i^x} \text{ if tasks are removed} \end{cases} \tag{3}$$

B. Medium Formalization

Since the medium is the main responsible for the transmission of the exchanged data, the formalization of its parameters is crucial. It is important to recall that the total number of mediums within a network is *N-1* since in our hypothesis a node can only communicate with two adjacent nodes. The medium that relates the two nodes $Node_x$ and $Node_y$ is represented by Med_{xy}.

In this approach the functionality of the medium is similar to a real-time processor. The transmission of packets is analog to the execution of tasks. Two types of data are transmitted : The reconfiguration scenarios and the information issued from the sensors. Therefore, when transmitted, the reconfiguration scenario *Reconf* and the sensed data *SD* are respectively divided into the packets RP_{xy}^j and SDP_{xy}^j that are supposed to be sent from $Node_x$ to $Node_y$. The representation of these elements is given by Equations (4) and (5).

$$Reconf = \{RP_{xy1}...RP_{xyj}\} \tag{4}$$

$$SD = \{SDP_{xy1}...SDP_{xyj}\} \tag{5}$$

where *j* is the number of times that the two types of data are sent. The latter mentioned packets are considered as real-time jobs having a unified time of transmission (analog to the time of execution) and a common period of occurrence. The computation of the transmission time depends on the DR_{xy} related to the medium Med_{xy} and the size of the packet whose the procedure of calculation is explained in details in Section IV-B. Let S^j be the packet size. The transmission time T^j is given by Equation (6).

$$T^j = \frac{S^j}{DR_{xy}} \tag{6}$$

Each packet is also characterized by a relative deadline DPa^j_{xy} specific for each single packet Pa^j_{xy}. It is supposed that a data composed of several packets, is required to be sent from $Node_x$ to $Node_y$ in a fixed amount of time and before a specific deadline denoted by D_{xy}. The values of the relative deadlines corresponding to the related packets are then given by Equation (7).

$$DPa^j_{xy} = \begin{cases} \frac{D_{xy}}{j} \text{if i=1} \\ (j-1) * \frac{D_{xy}}{j} \text{otherwise} \end{cases} \tag{7}$$

Moreover, priority P^i of each packet Pa^i_{xy} is the same as the related data (reconfiguration scenario or sensed data). It is to mention that the range of the priorities given to the reconfiguration requests is higher than the one given to the sensed data. In the case of an external intervention an exception can be made. The energy consumption due to the transmission of Pa^i_{xy}, denoted by ET^i, is also taken into account. In fact, this value depends on the transmission time of the latter. Let $cons(t)$ be a random function that describes the level of energy consumed at the moment t and $start_i$ be the moment at which the packet starts the transmission. Equation (8) gives the computation procedure that allows having the value of ET^i.

$$ET^i = \int_{start_i}^{start_i+T^i} cons(t)dt \tag{8}$$

III. Problem Modeling

When a reconfiguration scenario is expected to be sent to a specific node, a number of problems can occur. These problems can either be related to the node itself or to the communication between the nodes identified in the medium.

A. Node Related Problems

The main problems that a node can face after the application of several consecutive reconfiguration scenarios are : the exhaustion of the memory storage and the battery of the device as well as the non-respect of real-time constraints that should be satisfied by the tasks in the node. In fact, when a reconfiguration scenario adds tasks to be executed by $Node_x$, we can rapidly reach a moment t where the energy level $E_{C_x}(t)$ is null and/or no memory is available. This situation that describes the problem related to energy and memory exhaustion is given by Equation (9).

$$\exists\, t > 0,\ \forall x \in [1..N], Prob^E M_x : \begin{cases} M_x(t) = 0 \wedge E_{C_x}(t) = 0 \\ \vee \\ M_x(t) \neq 0 \wedge E_{C_x}(t) = 0 \\ \vee \\ M_x(t) = 0 \wedge E_{C_x}(t) \neq 0 \end{cases} \tag{9}$$

If we suppose that at a random moment t a scenario of reconfiguration is required to be applied to $Node_x$, the problem occurs when : the work left for the task takes time in a way that it exceeds its deadline or a task requires a resource used by another task or inversely. The real-time problem that takes place in a node is depicted by Equation (10) as follows.

$$\forall x \in [1..N]\,, \forall i, l \in [1, n_x]\,, Prob^{RT}_x : \begin{cases} \exists \tau^x_i / \tau^x_i.W > \tau^x_i.D - t \\ \exists \tau^x_i, \tau^x_l / \tau^x_i.Req(t) \cap \tau^x_l.Res(t) \neq \emptyset \\ \wedge \tau^x_l.Req(t) \cap \tau^x_i.Res(t) \neq \emptyset \end{cases} \tag{10}$$

B. Medium Related Problems

When dealing with the transmission of different types of data (reconfiguration scenarios and sensed data) having each a prefixed priority, a transmission deadline should not be exceeded. In fact, the energy related to the data transmission can be drained after a while due to the random communication of data from the current node to another one. When considering a medium Med_{xy} situated between two distinguished nodes $Node_x$ and $Node_y$, the mathematical representation of the energy loss related to transmission from $Node_x$ to the other within this medium is expressed by Equation (11).

$$\exists t > 0, Prob^{ET}_x :\ E_{T_x}(t) = 0 \tag{11}$$

Besides the energy problems, we can face the problem of exceeding the deadline of transmission when sending a specific type of data (either $Reconf_i$ or SD_i) from $Node_x$ to $Node_y$ which is caused by the confusion when choosing to send either the data having the highest priority first or the one having the least deadline. This problem is expressed by Equation (12).

$$Prob^{DT}_x :\ r_i + T^i > D_{xy} \tag{12}$$

where r_i refers to the release instant of the data to be sent.

Generally, in a wireless sensor network where the reconfiguration is taking place and the real-time constraints are imposed, the problems that can possibly occur are either the loss of energy or the exceeding of deadline within $Node_x$ or at the level of a medium relating this node to $Node_y$, which is expressed by Equation (13).

$$\forall x \in [1, N],\ Prob_x : \left\{ \begin{array}{c} \exists \tau_i^x / \tau_i^x.W > \tau_i^x.D - t \\ \exists \tau_i^x, \tau_j / \tau_i^x.Req(t) \cap \tau_j^x.Res(t) \neq \emptyset \\ \wedge \tau_j.Req(t) \cap \tau_i^x.Res(t) \neq \emptyset \\ \vee \\ E_{T_x}(t) = 0 \\ \vee \\ r_i + T^i > D_{xy} \\ \vee M_x(t) = 0 \wedge E_{C_x}(t) = 0 \\ \vee \\ M_x(t) \neq 0 \wedge E_{C_x}(t) = 0 \\ \vee \\ M_x(t) = 0 \wedge E_{C_x}(t) \neq 0 \end{array} \right. \tag{13}$$

Therefore, a feasible RWSN is the one where all the mediums and nodes are feasible in a way that none of the above mentioned problems can occur in any of them.

IV. Proposed Approach

The approach proposed in this paper aims to solve the problems previously mentioned. In fact, it offers a node based solution that allows treating the reconfiguration scenarios once they reach the latter, and a medium based solution allowing the circulation of these scenarios in coherence with sensed data transmission all along with an energy optimization and a required real-time feasibility.

A. Node Based Solution

The problems related to $Node_x$ ($x \in [1..N]$) and exposed in previous sections are solved in the current section.

1) Data Format: To encode the reconfiguration scenarios and sensed data in an appropriate way, they are both formatted into xml files. This allows the transferability of these files between heterogeneous devices, networks and systems sufficiently. Moreover, the parsing of these files into C code (language of most of the operating systems in WSN) makes it easier for the node to execute the received reconfiguration scenarios. Therefore, handling this type of format during programming a reconfiguration scenario should be easy and transparent for the developer located in another network as well as encouraging the interchangeability between different types of sensors and systems. Fig. 1 shows a possible scenario where sensed data are requested from the base station and a reconfiguration scenario is sent. In order to assure a format that is close enough to the formalized notions, common structures of files related to sensed data and the reconfiguration scenarios are proposed. As for sensed data, the basic information is the value received from a specific sensor which is properly converted. Therefore, the xml file related to the latter indicates the *id* and sensed values for each sensor in the node. This format is given by Fig. 2. As for the file related to reconfiguration scenarios, their description information as well as the related tasks are all mentioned. The parameters of the reconfigurations are first included in the file then those of each task. This format is given by Fig. 3.

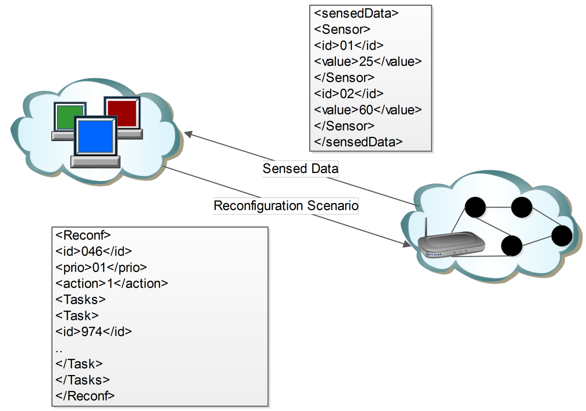

Fig. 1: Exchange of reconfiguration scenarios and sensed data between the node and the base station.

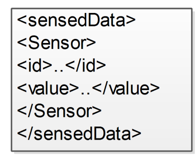

Fig. 2: XML format of the sensed data.

2) Node Structure: The proposition of a more optimal architecture of the sensor node is among the aims of the current approach. In addition to the inclusion of a pipelined middleware described in [16], the module responsible for the xml procedures and the one responsible for the network computations are also added. The structure of the proposed architecture is shown in Fig. 4.

These added modules are described as follows:

- **XML Unit**: This unit has two distinct roles. When receiving a reconfiguration scenario, it parses it into a C code to be treated by the pipelined middleware. However, when sending any sensed data, this unit receives the related information, arranges it into an xml file and handles it back to the operating system that is in charge of sending it to the concerned destination.
- **Network Computation Unit**: This unit takes into account the preparation of the sent data as well as the checking and reception of the received ones. In fact, when receiving a message, the node gathers the related packets and makes a whole file out of them. When the file represents a reconfiguration scenario, it gets converted into C code and treated in the pipelined middleware. Otherwise, it gets divided into packets again with different sizes. These packets are sent in a specific order all along with other types of data using an optimal scheduling such as the technique called PEDASA presented in [17]. This technique is described in details in Section IV-B.
- **Pipelined Middleware:** Each time a reconfiguration request is received, this middleware is activated. Based on the pipeline of reconfiguration, this middleware is responsible of running through the different modules in order to have an optimal execution of the node [16].

```
<Reconf>
<Rid>..</Rid>
<Priority>..</Priority>
<Deadline>..</Deadline>
<action>..</action>
<Tasks>
<Task>
<Tid>..</Tid>
<Sensor>
<Sid>..</Sid>
<Period></Period>
<TPriority>..</TPriority>
<TDeadline>..</TDeadline>
<Operation>
</Operation>
</Sensor>
</Task>
</Tasks>
</Reconf>
```

Fig. 3: XML format of the reconfiguration scenario.

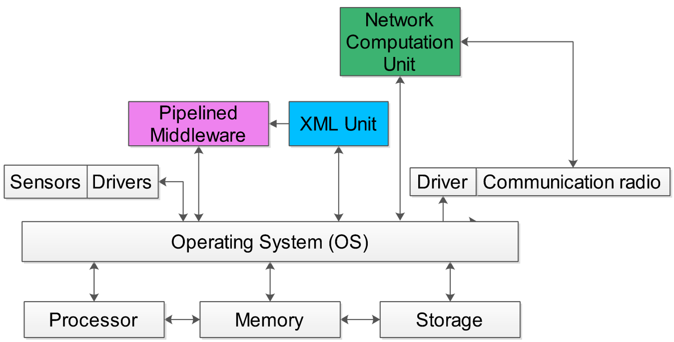

Fig. 4: Structure of the new node software architecture.

1) **System Checking Module:** In this first step it is essential to check the availability of the needed resources. Consequently, if a reconfiguration requests resources that are available, it is accepted. Therefore, the accepted reconfigurations, denoted by ρ_{SC}, is given as follows:

$$\begin{aligned}\rho_{SC}(t) = \{\exists Reconf_k | \forall \tau_i \in Reconf_k.\Gamma \\ \forall R_j \in \tau_i^{Res},\ R_j \in R(t)\}\end{aligned} \tag{14}$$

2) **Stability Module:** In this phase, the group of reconfiguration inquiries is organized in a queue to be treated by a decision making unit. This module is composed of two parts. In the first part there is a queue that contains the reconfiguration scenarios during a specified waiting time, named Δt. The second part is responsible of the decision making which consists of applying independent reconfigurations. The set of accepted reconfigurations in this module, denoted by ρ_S, is given as follows:

$$\begin{aligned}&\rho_S(t) = \{\exists Reconf_k \in \rho_{SC}(t) \\ &|(Reconf_k.\beta = \emptyset) \\ &\vee (Reconf_k.\theta = max_j(Reconf_k.\theta) \wedge Reconf_k.\eta = +1) \\ &\vee (Reconf_k.\theta = max_j(Reconf_k.\theta) \wedge Reconf_k.\eta = -1) \\ &\wedge (Reconf_k.\Gamma \subset \tau_{sys}(t))\}\end{aligned} \tag{15}$$

3) **Memory Module:** The control of the memory of $Node_x$ is realized in this module. At this level the tasks of the accepted reconfigurations are analyzed. Accepting the reconfigurations that remove tasks makes an important gain in terms of memory. Now it is possible to sort the reconfigurations that add tasks based on their priorities. The tasks existing in these reconfigurations are therefore sorted by using Rate Monotonic scheduler. The list of tasks accepted through this module is described by Equation (16).

$$\begin{aligned}&\rho_M(t) = \{\forall \tau_k \in \rho_S(t).\Gamma, \\ &Card(\tau_k) = Maximum | M_{\tau_k} < M(t)\}\end{aligned} \tag{16}$$

4) **Energy Module:** Since we assume that tasks have a precise time of execution and known real-time parameters, the energy consumed by each task is anticipated. An task that consumes energy should not be executed no matter what priority it has. The available levels of energy compared to the tasks are not taken into consideration in this case. A power factor $EF(t)$ represents an alternative to the energy level applied in the algorithm. One of the main purposes of this work is to extend the life cycle of the system and to prevent the exhaustion of the remaining energy before the moment of reloading. The final list refers to the maximum number of tasks that are able to operate in the system without exhausting the overall energy level as described by Equation (17).

$$\begin{aligned}&\rho_E(t) = \{\forall \tau_k \in \rho_M(t).\Gamma, \\ &Card(\tau_k) = Maximum | E_{\tau_k} < E(t)\}\end{aligned} \tag{17}$$

5) **Reconfigurable Priority Ceiling Protocol Module:** This protocol denoted by RPCP and fully described in [18] is a real-time protocol that avoids deadlocks after any reconfiguration scenario and changes the priorities of tasks in order to reduce their response and blocking times, and to meet their deadlines. This protocol requires the use of two virtual processors in order to guarantee the non-interruption of execution during the reconfiguration step. This corresponding module takes in charge changing the priorities that belong to the resulting tasks coming from the previous module (Energy module).

$$\rho_R : \begin{cases} \forall \tau_i, \forall \tau_j | \tau_i \neq \tau_j \\ \wedge \tau_i^{Req(t)} \cap \tau_j^{Rest(t)} = \emptyset \\ \forall \tau_i | \tau_i^B = Minimum(\tau_i^B) \\ \forall \tau_i | \tau_i^R = Minimum(\tau_i^R) \\ \forall \tau_i | \tau_i^W < \tau_i^{D-t_1} \\ \forall \tau_i | \sum_{i=1}^{n} \frac{Ci}{Ti} \leq n * (2^{\frac{1}{n}} - 1) \end{cases} \tag{18}$$

B. Medium based Solution

Just like giving the solution to $Node_x$ in the previous section, the proposed approach related to medium Med_{xy} which connects the node $Node_x$ to $Node_y$ is given in the current section.

1) Network design: In order to resolve network related problems described by Equations (11) and (12), a network design described by Fig. 5 is proposed. This design considers *Sender* and *Receiver* nodes joined by *Transmission Medium*. At the *Sender* level, three stages are faced: The application

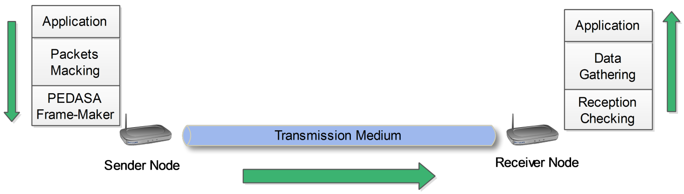

Fig. 5: Structure of the new node Network architecture.

where the software procedures are conducted, the *Packet Making* part and the PEDASA *Frame Maker*. In fact, as previously mentioned, in order to be able to send the whole files from a source to a destination, it is necessary to decompose them into packets having specific sizes. Therefore, the role of the *Packet Making* part is to assure this constraint. At this level, the decision concerning the data to be sent is already made. The next step consists in decomposing this data in a way where each file is split into the same number of packets as others. This allows constructing frames that have the exact type of information repetitively. Each frame contains a single packet from each sent data file. The size of the frames should not in any way exceed the bandwidth of the medium. This is given by Equation (19).

$$\sum_{i=1}^{z}\sum_{j=1}^{k} Size(SD_j^{xy}) < DR_{xy} \tag{19}$$

where z is the number of files to be sent and *k* is the number of packets. The value of the latter is determined by Algorithm 1. Therefore the size of each packet is determined by dividing the size of the file by the number *k* as shown in Equation (20).

$$\forall i \in [1..k] \;\; S^i = \frac{Size(SD)}{k} \tag{20}$$

Once the files are decomposed into packets, it comes the time

Algorithm 1 Computation of the number of packets per frame.

Data: List of sent data
Result: k
$k \leftarrow 0$
$SizeFrame \leftarrow 0$
while $SizeFrame < Bandwidth + 1$ **do**
  $k \leftarrow k + 1$
  for $j = 1; j < Size(sentdata); j++$ **do**
    $SizeFrame \leftarrow +Round(sentdata[j]/k)$
  end
end

of constructing the frames. In fact, by frame we refer to the order at which the packets are sent from $Node_x$ to $Node_y$ by crossing Med_{xy}. As mentioned in Section II-B, each packet (independently from its type) has priority P^i, relative deadline $Derel_i$ and energy consumption ET^i related to the transmission. The order of transmission that guarantees the respect of these parameters is realized by means of PEDASA Frame Maker.

In fact, PEDASA [17] is a static algorithm that consists on manipulating certain parameters in order to allow all of real-time tasks to meet their deadlines and to be executed before the battery exhaustion according to their static priorities. In this part, the packets are treated as virtual real-time tasks which allow the application of the mentioned algorithm. The procedure of computing the new priorities is based on three factors, α_{xy}^P, α_{xy}^D and α_{xy}^E, related to the importance of respectively: The existing priority, how close the deadline is and how low the consumed energy is. The relation between these rates is given by Equation (21).

$$0 \leq \alpha_{xy}^P, \alpha_{xy}^D, \alpha_{xy}^E \leq 1, \alpha_{xy}^P + \alpha_{xy}^D + \alpha_{xy}^E = 1 \tag{21}$$

The packets are then sorted considering an ascending order for the fixed priority and a descendant one for deadline and energy consumption. The idea behind this calculation is to have resultant priorities related to the packets crossing Med_{xy} and issued from the existing priorities (Equation (22)). Instead of running through all the possibilities, we focus on the preferences given by the system at first place. This guarantees the respect of the first desired parameters.

$$K_{xy} = \begin{pmatrix} \alpha_{xy}^P \\ \alpha_{xy}^D \\ \alpha_{xy}^E \end{pmatrix} * M_{xy\alpha} \tag{22}$$

where $M_{xy\alpha}$ is the matrix relating the order factors of the tasks to the defined parameters. This matrix, having as dimensions n_x rows (the number of tasks) and 3 columns (corresponds to the system parameters), is composed of the elements $m_{xy\alpha ij}$. Using the obtained equations relating the three factors to the new priorities, it is possible to run a heuristic study that allows having the final values of these priorities. Accordingly, the main purpose of this approach is to maximize the number of packets respecting their deadlines without totally ignoring the delimited priorities. In fact, the predefined priority set does not allow all of the packets to fully respect their deadlines. Hence, we aim for less than the totality of packets respecting the desired arrangement. While searching for the new set of packets, we consider the fact that: **(i)** When fixing the importance factors, we should have a number of packets respecting their deadlines that is greater or equal to the number of packets whose old priorities correspond to the new ones, and **(ii)** The multiplication of these two numbers should be the maximum amongst all the possible values. The choice behind this analysis is founded on the desire of guaranteeing, so far, a higher number of packets respecting their deadlines while partially obeying the predefined priorities. However, the complication occurs once the number of deadline-conducted packets after calculation is lower than the one initially obtained by the fixed priority preemptive algorithm [29]. in this way, we focus on the set of priorities offered by this scheduling algorithm. Nevertheless, as long as there are packets exceeding their deadlines, we proceed to change the periods of the latter ones in order to have a fully functional system. The fundamental intention here consists on enhancing the values of a minimum number of periods related to the least prioritized packets which outstrip their deadlines and replacing them by their multiples. Yet, the incrementation should obey to the constraint that we should not outpace the existing hyper-period.

After finding the correct order in which the packets should be sent, the concept of the system reloading is then introduced. In fact, instead of manipulating the parameters of the set of packets, it seems more efficient to reload the energy within the system, at a specific moment, in a way that its energy level gets restored. That is why in this part we search for the instants of energy reloading that enhances the lifespan of the whole system. Obviously, between the reloading moments, the energy level can either decrease or remain the same. This depends on the energy required by each packet. The chance of rewinding the system allows the latter to send all of its packets without worrying about the exhaustion of the battery. We proceed in a deterministic way since the proposed scheduling algorithm is static. The first step consists on determining the instants at which each packet gets preempted by another of a higher priority and the ones at which it resumes its transmission. The second step takes in charge the definition of the new consumption functions that result from the several preemption cases that a packet may have. Supposing that the procedure of reloading the battery is immediate and that the time it takes is insignificant, the reloading moment should more likely occur when the general consumption function attains zero. Although this assumption is unrealistic, its use is mainly a measure of simplification. As a consequence, the system can dispose of a set of parameters ready for application without worrying about any future behavior since the periodicity is always predictable. The result is then a set of instants at which the transmission battery should be reloaded and a set of new priorities. Equation (23) describes the result on the PEDASA computation.

$$\forall i, j \in [1..n]\, PEDASA(SD_{ij}) = \big(ReloadM, New\_P^i\big) \tag{23}$$

where *ReloadM* is the vector containing the reloading moments of the transmission battery and New_P^i refers to the set of new priorities.

In the side of the *Receiver* level, the *Reception Checking* and *Data Gathering* units are important to assure an effective reception of data files. In fact, the first unit is responsible for checking both the correctness of the received frames and the intended destination. As for Data Gathering unit, its role is to form the files out of the received packets if they are accepted by the previous unit.

2) Communication Protocol: The steps that the two nodes $Node_x$ and $Node_y$ follow in order to send and receive scenarios of reconfiguration as well as sensed data are described by the proposed communication protocol as follows:

1) Files of sensed data and scenarios of reconfiguration are ready to be sent from *Sender* $Node_x$ to *Receiver* $Node_y$.
2) The Packet Making Unit in $Node_x$ decomposes the files into packets having specific sizes. To each packet the destination address as well as the parameters defined in section II-B are associated.
3) The PEDASA computation is then performed in order to create the frames.
4) A first *model frame* is sent from $Node_x$ to $Node_y$ indicating the number of faced transmission and the followed pattern (the order of the packets).
5) At *Reception Checking* level, $Node_y$ checks the available energy and confirms the receiving procedure by sending back an acknowledgment message to the *Sender* $Node_x$.
6) $Node_x$ starts sending the frames. It waits for an amount of time after each sent frame. If there is any refusal message, then it sends back again.
7) The *Reception Checking* unit collects the received data and redirects it to *Data Gathering* unit if $Node_y$ is its reception address. At this level, the latter mentioned unit assembles the packets into xml files to be proceeded to the pipeline. If the target of the packet is another destination, $Node_y$ behaves as a sender node and repeats the mentioned procedures.

C. Feasibility of RWSN

The gain at the level of a single node impacts the overall network positively. In fact, through this approach each node is allowed to gain in terms of both memory and processing energy thanks to the pipeline of reconfiguration. Moreover, each related task is executing under respect of real-time constraints. The application of PEDASA premise also benefits the nodes from the expansion of their lifespan due to conserving both the transmission and reception energy levels. The reflection during the packet making procedures allows the respect of the imposed priorities of the exchanged data as well as the real-time constraints related to their transmission. Therefore, the whole RWSN is considered to be feasible if every node composing it satisfies these conditions given by Equation (24).

$$\forall x, y \in [1, N], \forall Node_x \in RWSN: \\ \begin{cases} \forall t > 0, M_x(t) \neq 0 \\ \forall t > 0, E_{C_x}(t) \neq 0 \\ \forall t > 0, E_{T_x}(t) \neq 0 \\ \forall t > 0, E_{R_x}(t) \neq 0 \\ \forall \tau_i^x, i \in [1, n_x], A_i^x + R_i^x < D_i^x \\ \forall Pa_{xy}^{kj}, j \in \left[1, nP_{xy}^k\right], k \in [1, nD_{xy}] \\ T^{kj} < DPa_{xy}^{kj} \\ \sum_{j=1}^{nP_{xy}^k} T^{kj} < D_{xy}^k \end{cases} \tag{24}$$

In order to expose the complexity of the proposed approach, Let m_{rs} be the number of reconfiguration scenarios to be applied in RWSN, n_{tk} be the average number of tasks to be executed in each node after any reconfiguration scenario, and n_{pt} be the average number of messages to be exchanged after any reconfiguration scenario. The problem complexity is given by

- Computation complexity in Nodes based on the pipeline oriented approach:
  - The complexity of the stability module, allowing the management of concurrent reconfiguration scenarios, is $O(n_{tk}.log(n_{tk}))$,

- The complexity of the memory and energy modules is O(1),
- The complexity of the RPCP module is $O(n_{tk}.log(n_{tk}))$,

- Computation complexity of PEDASA algorithm to sort n_{pt} packets according to their real-time, energy and static priorities is $O(n_{pt}.log(n_{pt}))$,
- The complexity of the proposed approach, where m_{rs} reconfiguration scenarios are considered, is is:

$$O(m_{rs}.(n_{tk}.log(n_{tk}) + n_{pt}.log(n_{pt}))). \quad (25)$$

V. Case Study

The proposed approach is applied to a real case study in order to pin point the gains offered from this proposition. This premise is applied to a WSN used for the automobile journey conducted as an experiment by the embedded systems department in the company ARDIA.

A. Presentation

An initial application of the wireless sensor network is conceived in order to measure, process and supply diverse types of information during an automobile journey. Examples are acceleration and fuel consumption, identification of incorrect tire pressure, verification of illumination, and evaluation of the vital signals of the driver. In each car, several nodes are implemented. Each node contains five types of sensors:

- **Air Flow Sensor:** a device that measures the quantity of air going into the internal combustion engine,
- **Tire Pressure Sensor:** this device indicates how much load each of the four tires is carrying,
- **Light Sensor:** the role of this device is to verify the state of the automobile lights,
- **Acceleration Sensor:** this device is responsible for checking the accelerator of the vehicle,
- **Temperature Sensor:** this device is installed in contact with the engine in order to monitor its temperature.

For a precise case study, we consider three cars each having three nodes (from the set of several cars present in the automobile journey). The procedure is realized by an operator representing the base station. The whole set serves as a complete wireless sensor network where the nodes interact with each other in order to convey information from the base station to the nodes reversely. This information can either be simple data extracted by the sensors in the nodes or reconfiguration scenarios sent to the latter in order to adapt their behavior to the specified requirements.

At the level of a single node, we consider a set of reconfiguration scenarios that are performed during a whole month (30 days). Each day, the number of requested reconfigurations can alternate from 1 to 3. The number of tasks per reconfiguration also varies between 1 and 3. The main tasks are related to the sensors, and each task is capable of consuming a specific amount of both energy and memory as described by Table I. This amount is computed as a predefined percentage from the total of both available memory and energy within the node.

Tasks	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
Energy Consumption	15	23	30	28	24	16	10	21
Memory Consumption	20	24	19	16	15	29	17	15

TABLE I: List of possible tasks.

On the other hand, a different node plays the role of a switch, conveying reconfiguration scenarios from the base station to other nodes and data from a different node back to the base station. We consider that at the level of a specific node, a number of scenarios of reconfigurations as well as sensed data are ready to be sent to another node. This set of information is given by Table II.

Sent Data	Deadline(s)	Priority	Size(Kbit)
Reconf1	20	4	1000
Reconf2	30	5	1500
SD1	40	2	300
SD2	10	3	200
SD3	25	1	350

TABLE II: Details of the sent Data.

The chosen medium is based on the technology ZigBee [30] which offers a rate of 250Kbit/s. Moreover, each of the transmitted data is characterized by its eventual deadline in seconds, its given priority and its size in Kbit.

B. Problem

During the functioning of the system previously presented, many anomalies are deduced during the several tests conducted in ARDIA. Considering the level of a single node, the first problem concluded while executing the consecutive reconfigurations on the node is the quick loss of energy. In fact, by Day 2, no energy is left in the battery to execute the rest of the tasks and apply what remains of the needed reconfiguration scenarios. For several times (16 times) the level of the available energy decreases dramatically during the application of the reconfigurations, preventing 13% of the reconfiguration scenarios from being executed. The exhaustion of the memory makes no exception. In fact, the memory level gets overloaded several times as well. Figures 10 and 11 give an overview on the state of evolution of the available power and memory during the application of the different reconfiguration scenarios. Moreover, the real-time aspects during the execution of the tasks are also impacted by the reconfiguration scenarios. It is possible to remark that adding the new task τ_7 during the execution of the system effects the deadlines of the existing tasks.

Likewise, when considering the transmission of the data, other problems are observed. Namely, the loss of energy due to the large amounts of transmitted data and the violation of the real-time constraints related to reception of data before the specified deadline are the main problems. In fact, during the transmission of the specified packets related to the reconfiguration scenarios and the sensed data, the energy level reaches a value that does not allow the transmission of the SD3. As shown in Fig. 6, although the energy level does not attain 0 yet, it does not allow the transmission of the rest of the required packets.

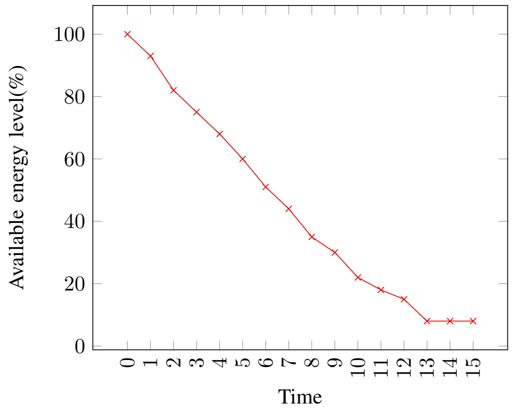

Fig. 6: Fatal decrease in energy level during transmission.

In this given case, it is possible to remark the violation of the data SD2 when the elements are transmitted according to their priorities.

C. Implementation

The implementation of the solution consists, mainly, in defining a new hardware design of the node itself and a software resolution of this proposition. The latter mentioned design, shown by Fig. 7, is based on a supply circuit allowing the dynamic reloading of the node using a solar panel technique.

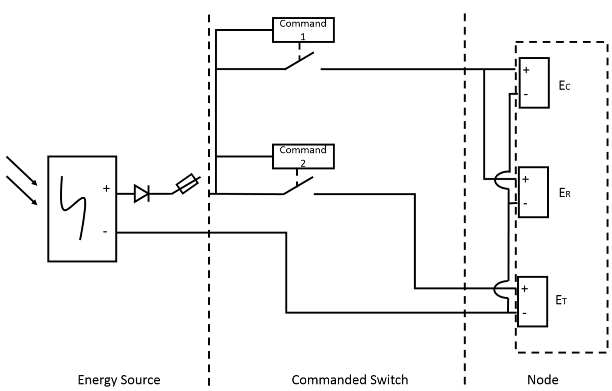

Fig. 7: Structure of the new node hardware architecture.

This architecture is composed of three parts: Energy Source, Commanded Switch and the Node. The solar panel in the first part, considered as an infinite source of power, is followed by a diode preventing the return of the electrical current to the source and a disconnect fuse holder protecting the whole circuit from any overload or short-circuit. At the level of the node, we distinguish three batteries each supplying the transmission, reception and computation units. Every battery is stoked in a commanded way represented by the commanded switch part. In fact, both the reception and the computation batteries are supplied periodically. That is why the switch that connects them to the source is controlled by a smart device (*Command1*) that links them at a specific instant. As for the battery related to the transmission, the instants at which it should be reloaded are determined by the PEDASA unit. The instants are then communicated to *Command2* that knows exactly at which moments the operation needs to be performed. The solar panel is the main source of energy in this node. The switches are controlled by micro-controllers responsible of indicating the right moments at which the batteries should be charged. The first micro-controller uC1 takes in charge the periods at which the switch related to charging the battery of the reception and the one of computation. On the other hand, uC2 stocks a table containing the right moments extracted from the PEDASA computation to charge the battery related to the reception unit. Fig. 8 exposes the moments of any fatal decrease.

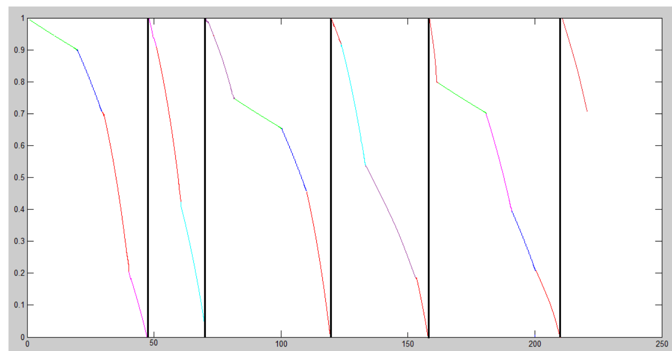

Fig. 8: Moments of reloading concluded from the moments of decrease of energy levels.

When it comes to the core of the node, two boards are used: the first takes care of the reception and the pipeline procedure as well as the sensing, while the second makes the PEDASA calculations and the reception of data. It is worth to mention that the arduino boards are both connected and exchange the necessary information with the previously mentioned micro-controllers too. As for the PEDASA calculations, they are allowed to send the packets over 14 superframes. Initially, a portion of each datum given in Table II is sent. Therefore, the format of the superframe is as shown in Fig. 9. Moreover, by time the number of packets decreases allowing more data to be sent.

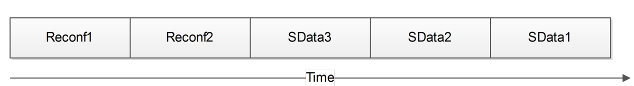

Fig. 9: Initial Superframe sent after PEDASA Calculation.

D. Discussion and comparison with other approaches

The performance of this work consists in its capability in the resolution of the real-time problems as well as the conservation of the memory and the energy levels that guarantee the extension of the time span of the whole system thanks to the pipeline of reconfiguration. As remarked in Figs. 10 and 11, most of the tasks benefit from the available energy before the

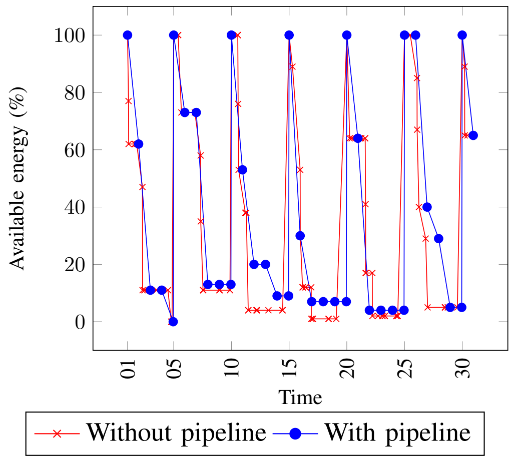

Fig. 10: Evolution of the available energy E_C through time with and without pipeline in a single node.

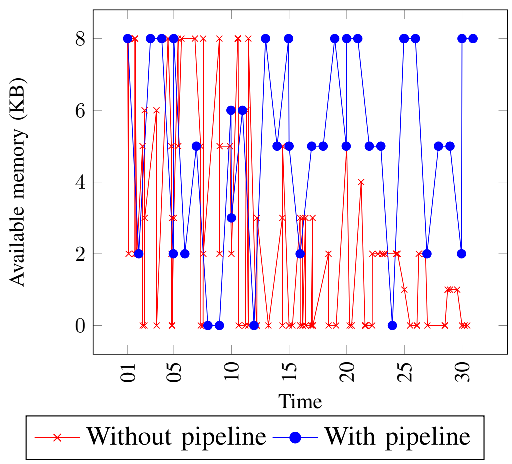

Fig. 11: Evolution of the available memory through time with and without pipeline in a single node.

following reloading moment. This allows a bigger number of tasks to be applied to the system comparing with the case of not employing the pipeline. Indeed, the rate of lost tasks before using the pipeline is around 25%. It is reduced by using the proposed solution to only 20%. The conservation of both the energy and the memory at the level of the node forms a major original contribution of the proposed approach.

Moreover, the PEDASA procedure plays also an important role in maintaining an effective transmission of data. This procedure allows the flow of data from different types and priorities to be transmitted without losing energy and respecting the real-time criteria. In fact, this technique is not the only one in the literature that treats the static formation of data frame based on prioritizing the elements. **Approach 1** [20] takes into consideration pinpointing a specific approach for network flows scheduling in home automation applications based on IEEE 802.15.4 wireless sensor networks. It introduces rate monotonic (RM) for time slots allocation combined with priority-based CSMA/CA. In **Approach 2** [19] a real-time wireless media access control protocol based on earliest

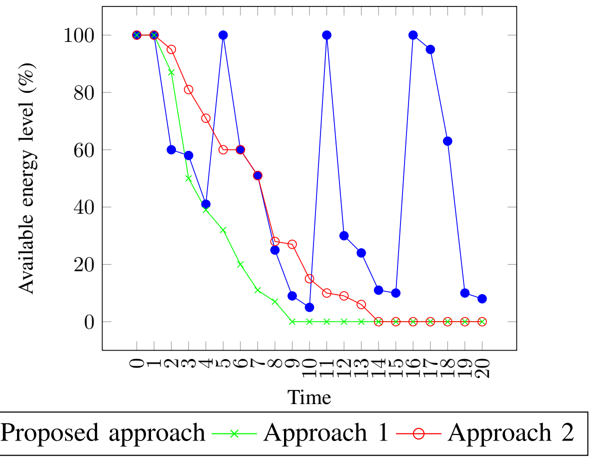

Fig. 12: Graph of comparison between the proposed approach and the other works in terms of energy.

deadline first scheduling (EDF) scheme is proposed as well. The contribution of PEDASA is significant comparing with RM in **Approach 1** and with EDF in **Approach 2**. In fact, EDF and RM only consider the scheduling of tasks based on their deadlines as any mono-criteria scheduling algorithms. On the other hand, PEDASA makes a compromise between the energy consumption and the initial functional priorities. The aim of this strategy is to relate the new set of priorities to the first given parameters of the three criteria as well. The output of this algorithm is not only the new priorities, but also a new definition to the periods of the least prioritized tasks in a way that guarantees the feasibility of the scheduled task set. Moreover, the instants of reloading relevant to the battery are also resulted from the PEDASA algorithm.

Comparing with the approach proposed in this paper, the network lifespan in the related works is relatively reduced, which stems from the fatal decrease of energy through time. In fact, one of the main originality offered by the proposed approach is the capability of the nodes to reload their energy levels. This feature, based on a range of computations, extends the interoperability of the nodes and prevents the exhaustive search for a backup plan.

Moreover, the proposed approach offers the particularity of treating the reconfiguration scenarios differently in a way that assures their arrival to the specified destination before their deadlines. However, the related works do not give any special interest to these important scenarios.

In order to expose the gains in terms of energy related to both reception and transmission of packets, we compute the values of these gains while varying both the time and the number of transmitted packets. Figs. 13 and 14 show the variation of the transmission and reception energies E_T and E_R relatively. It is obvious that these levels decrease dramatically by time especially when the number of packets is important. They even reach zero within a short time period leaving the system in a fatal paralyze. However, the use of the reloading method introduced by PEDASA prevents the aforementioned energy levels to remain available and to never reach the bottom of their values even when the number of

Work	Reconfiguration Strategy	Energy management Strategy	Real-time management Strategy
EEDF-MAC [19]	unsupported	The energy efficiency is achieved by turning off the radio on the nodes which are neither in transmission nor in reception state.	Based on EDF for transmission.
[20]	unsupported	unsupported	Based on RM for transmission.
Agilla [36]	It supports adaptive applications through mobile agents that coordinate via localized tuple spaces.	The agents perform basic tasks like obtaining the neighbor list, sensing, periodically sleeping to conserve energy performed.	unsupported
[35]	It presents a modular architecture used for control applications in wireless sensor/actuator/controller networks that allows component integration and system reconfiguration at run-time.	unsupported	For the computation schedulability analysis the authors use standard real-time response analysis and the mode-change protocol adapted for the Embedded Virtual Machine.
Proposed Approach	The reconfiguration scenarios are sent from the base station to the nodes.	The nodes use a table containing the right moments extracted from the PEDASA computation to charge the batteries related to the reception, transmission and computation.	Transmission and computation are respectively based on PEDASA and RPCP.

TABLE III: Comparative Study

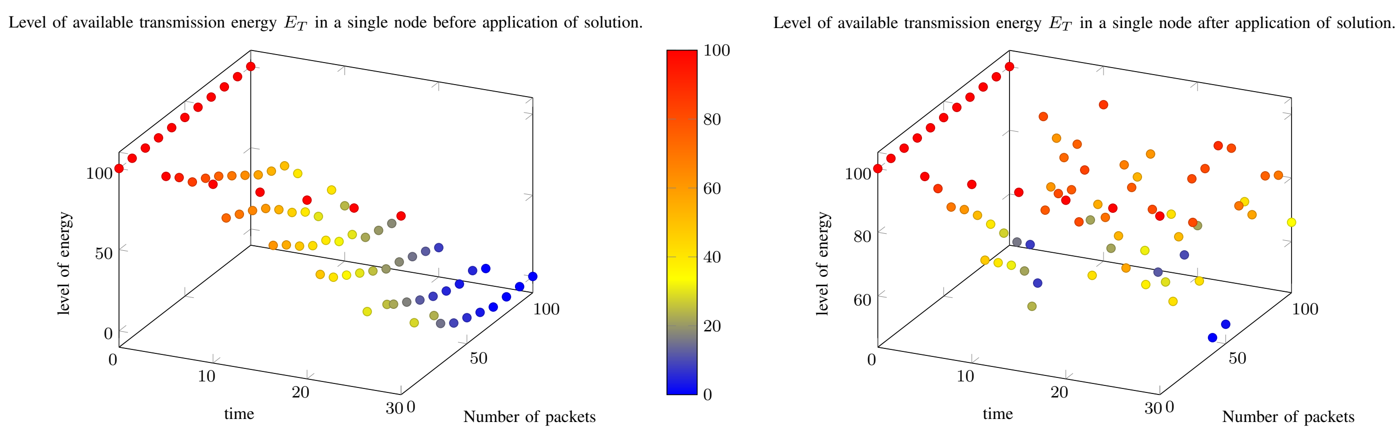

Fig. 13: Level of available transmission energy E_T in a single node before and after application of solution while varying time and number of reconfiguration scenarios.

packets increases. Therefore, it is possible to notice that both E_T and E_R are maintained stable during the experiment.

Table III compares the developed approach in this paper with the related works in the literature. The discussed approach analyzes all the existing alternatives and chooses the most useful one to resolve the encountered problem.

VI. Conclusion

The proposed approach does not only focus on the level of a single node, but also takes into account the totality of the network. Thanks to the pipeline of reconfiguration, the decision making problems that result from the successive requests of reconfiguration are resolved. The resulting device is both memory and energy aware and the acknowledgment of available sensors makes the decision making easier. In fact, having several steps in the procedure of receiving the reconfiguration request makes it possible to choose the one to be applied based on the stability module algorithm. Besides of that, both the memory and the energy levels are regulated to allow the node to accept as more tasks as possible. Moreover, considering the medium as a real-time processor makes the scheduling of data during transmission more effective and sufficient. In fact, PEDASA offers the possibility of the multi-criteria decision making in a computational way. Similarly, the establishment of power reloading possibility in predetermined instants guarantees the continuous functioning of the system. This work takes into account predictive reconfiguration scenarios which are known by the nodes as well as the related events. As a perspective, we will be interested in handling also non-predictive scenarios by remotely uploading any required task code in the target nodes. Moreover, we will be interested in optimizing the codes of reconfigurable tasks as well as the required synchronization between the nodes in coordination.

References

[1] F. Silva, “Industrial wireless sensor networks: Applications, protocols, and standards”, *IEEE Trans. Ind. Electron. Magazine*, vol. 8, no. 4, pp. 67–68, 2014.

[2] A. Pathak and V. Sarkari, “Wireless sensor network”, *Asian J. of Comput. Sci. and Inform. Technol.*, vol. 4, no. 7, pp. 67–71, 2014.

[3] B. S. Faical, F. G. Costa, G. Pessin, J. Ueyama, H. Freitas, A. Colombo, P. H. Fini, L. Villas, F. S. Osrio, P. A. Vargas and T. Braun, “The use of unmanned aerial vehicles and wireless sensor networks for spraying pesticides”, *J. Syst. Architecture*, vol. 60, no. 4, pp. 393–404, 2014.

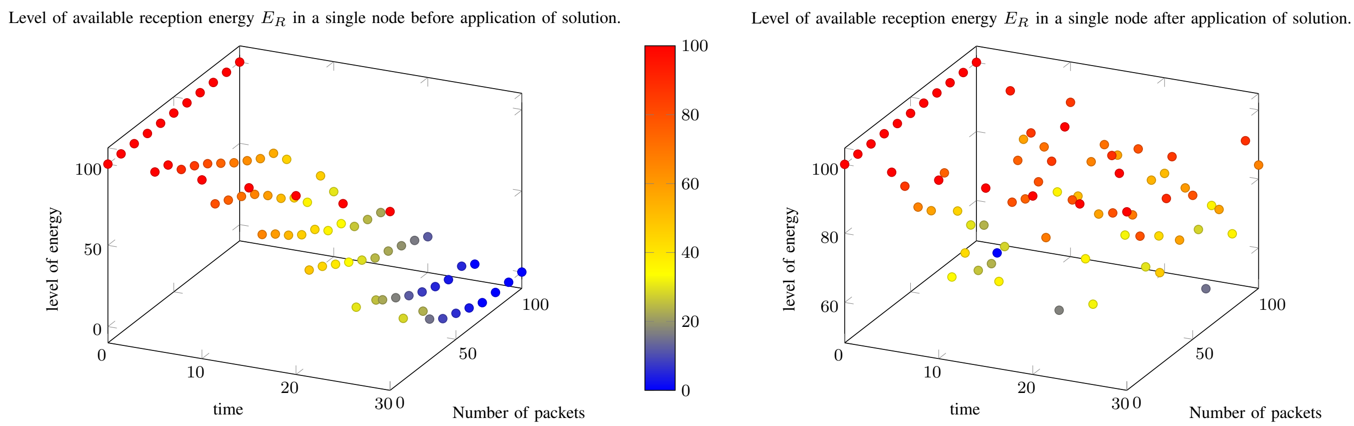

Fig. 14: Level of available reception energy E_R in a single node before and after application of solution while varying time and number of reconfiguration scenarios.

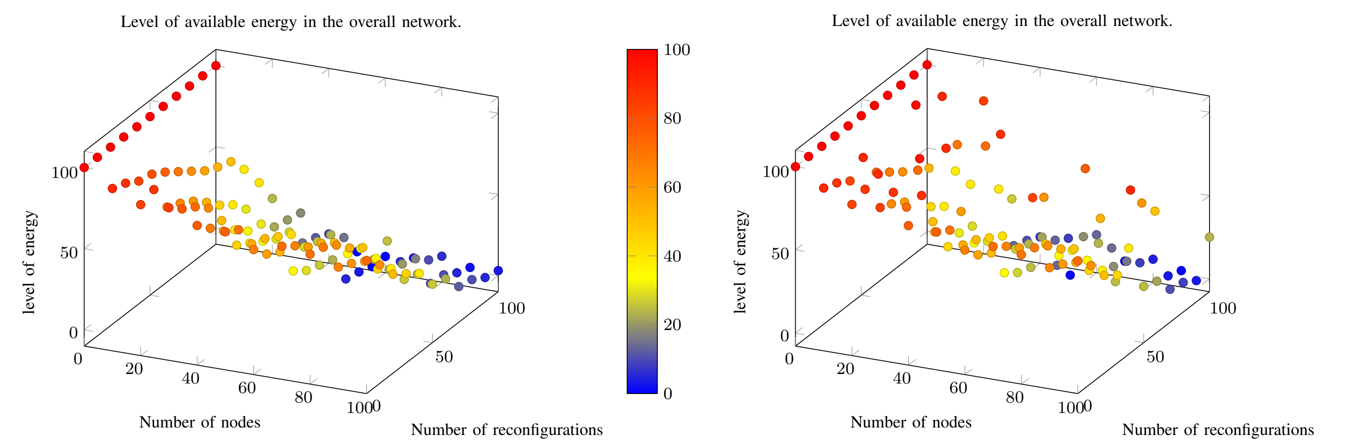

Fig. 15: Level of available energy in the network in general before and after application of solution while varying number of nodes and number of reconfiguration scenarios.

[4] D. J. Hatley and I. A. Pirbhai, *Strategies for real-time system specification*, Addison-Wesley, New York, NY, USA, 2013.

[5] R. Mancuso, R. Dudko, E. Betti, M. Cesati, M. Caccamo and R. Pellizzoni, "Real-time cache management framework for multi-core architectures", in *Proc. IEEE Real-Time and Embedded Technol. and Applicat. Symp.*, Philadelphia, PA, USA, pp. 45–54, 2013.

[6] L. Mo, P. You, X. Cao, Y. Q. Song and J. Chen, "Decentralized multi-charger coordination for wireless rechargeable sensor networks", in *Proc. IEEE Int. Performance Computing and Commun. Conf.*, Naning, China, 2015.

[7] D. C. Hoang, P. Yadav, R. Kumar and S. K. Panda, "Real-time implementation of a harmony search algorithm-based clustering protocol for energy-efficient wireless sensor networks", *IEEE Trans. Ind. Informatics*, vol. 10, no. 1, pp. 774–783, 2014.

[8] A. H. Dehwah, S. B. Taieb, J. S. Shamma and C. G. Claudel, "Decentralized energy and power estimation in solar-powered wireless sensor networks", in *Proc. Int. Conf. on Distributed Computing in Sensor Systems*, Fortaleza, Brazil, pp. 199–200, 2015.

[9] L. Fu, L. He, P. Cheng, Y. Gu, J. Pan and J. Chen, "ESync: Energy synchronized mobile charging in rechargeable wireless sensor networks", in *Proc. ACM Int. Symp. on Mobile Ad Hoc Netw. and Computing*, New York, NY, USA, pp. 247–256, 2015.

[10] M. Chen, "Reconfiguration of sustainable thermoelectric generation using wireless sensor network", *IEEE Trans. Ind. Electron.*, vol. 61, no. 6, pp. 2776–2783, 2014.

[11] L. Pezzarossa, M. Schoeberl and J. Sparso, "Towards utilizing reconfigurable shared resources in multi-core hard real-time systems", in *Proc. Junior Researcher Workshop on Real-Time Computing*, Lille, France, p. 21, 2015.

[12] S.W. Cheng, D. Garlan and B. Schmerl, "RAIDE for engineering architecture-based self-adaptive systems", in *Proc. Int. Conf. Softw. Eng.-Companion*, British Columbia, Canada, 2009.

[13] M. Castaldi, A. Carzaniga, P. Inverardi and A. L. Wolf, "A lightweight infrastructure for reconfiguring applications", *Software Configuration Manage., Lecture Notes in Comput. Sci.*, Springer, New York, NY, vol. 2649, pp. 231–244, 2003.

[14] S. Mitchell, H. Naguib, G. Coulouris and T. Kindberg, "A QoS support framework for dynamically reconfigurable multimedia applications", *Distributed Applicat. and Interoperable Syst. II, Int. Federation for Inform. Process.*, Springer, New York, NY, vol. 15, pp. 17–30, 1999.

[15] V. Kawadia and PR. Kumar, "Power control and clustering in ad hoc networks", *Annu. Joint Conf. of the IEEE Comput. and Commun.*, San Francisco, USA, vol. 1, pp. 459–469, 2003.

[16] M. Gasmi, O. Mosbahi, M. Khalgui, L. Gomes and Zhiwu Li, "R-Node: New pipelined approach for an effective reconfigurable wireless sensor node", *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. PP, no. 99, pp.1–14, 2016.

[17] M. Gasmi, O. Mosbahi, M. Khalgui and L. Gomes, "PEDASA: Priority, energy and deadline aware scheduling algorithm", in *Proc. Conf. Intelligent Software Methodologies, Tools and Techniques*, Naples, Italy, vol. 532, pp. 59–72, 2015.

[18] M. Gasmi, O. Mosbahi, M. Khalgui and L. Gomes, "Reconfigurable priority ceiling protocol: A safe way to real-time reconfiguration", *Informatics in Control, Automation and Robotics, Lecture Notes in Elect. Eng.*, Springer, Switzerland, vol. 370, pp. 23–42, 2016.

[19] M. M. Mohiuddin, I. Adithyan and P. Rajalakshmi, "EEDF-MAC: An energy efficient MAC protocol for wireless sensor networks", in *Proc. Int. Conf. on Advances in Computing, Commun.and Informatics*, Mysore, India, pp. 1323–1329, 2013.

[20] M. Collotta, G. Scat and G. Pau, "A priority-based CSMA/CA mecha-

nism to support deadline-aware scheduling in home automation applications using IEEE 802.15.4", *Int. J. Distributed Sensor Networks*, vol. 2013, p. 12, 2013.

[21] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks", *IEEE J. Selected Areas in Commun.*, vol. 17, no. 8, pp. 1333–1344, 1999.

[22] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment", in *Proc. Annu. Joint Conf. of the IEEE Computer and Commun. Societies*, Tel Aviv, vol. 2, pp. 404–413, 2000.

[23] L. Li, J. Y. Halpern, P. Bahl, YM. Wang and R. Wattenhofer, "Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks", in *Proc. Annu. ACM Symp. on Principles of distributed Computing*, Newport, Rhode Island, pp. 264–273, 2001.

[24] A. Cerpa and D. Estrin, "ASCENT: Adaptive self-configuring sensor networks topologies", *IEEE Trans. Mobile Computing*, vol. 3, no. 3, pp. 272–285, 2004.

[25] B. Chen, K. Jamieson, H. Balakrishnan and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks", *Wireless Networks*, vol. 8, no. 5, pp. 481–494, 2002.

[26] Y. Xu, J. Heidemann and D. Estrin, "Geography-informed energy conservation for ad-hoc routing", in *Proc. Annu. Int. Conf. on Mobile Computing and Networking*, Rome, Italy, pp. 70–84, 2001.

[27] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks", in *Proc. Int. Conf. on Embedded Networked Sensor Syst.*, Los angeles, CA, USA, pp. 28–39, 2003.

[28] T. Moscibroda and R. Wattenhofer, "Maximizing the lifetime of dominating sets", in *Proc. IEEE Int. Parallel and Distributed Processing Symp.*, Denver, CO, USA, p. 8, 2005.

[29] R. Li, B. Ma and R. Li, "Bounding the worst-case execution time for the fixed-priority preemptive systems based on the preemption Points", *Advances in Inform. Sci. and Service Sci.*, vol. 5, no. 10, p. 947, 2013.

[30] B. Qiao and K. Ma, "An enhancement of the ZigBee wireless sensor network using bluetooth for industrial field measurement", in *Proc. IEEE MTT-S Int. Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applicat.*, Suzhou, China, pp. 1–3, 2015.

[31] J. A. Nazabal, P. L.Iturri, L. Azpilicueta, F. Falcone and C. Fernndez-Valdivielso, "Performance analysis of IEEE 802.15. 4 compliant wireless devices for heterogeneous indoor home automation environments.", *Int. J. Antennas and Propagation*, vol. 2012, pp. 1–14, 2012.

[32] T. C. Minh, B. Bellalta and M. Oliver, "DISON: A self-organizing network management framework for wireless sensor networks", *Lecture Notes Inst. for Comput. Sci., Social Informatics and Telecommun. Eng.*, vol. 111, pp. 149–163, 2013.

[33] A. Jemal and R. B. Halima, "A QoS-driven self-adaptive architecture for wireless sensor networks", in *Proc. IEEE 22nd Int. Workshop Enabling Technologies: Infrastructure for Collaborative Enterprises* , Hammamet, Tunisia, pp.125–130, 2013.

[34] T. Bourdenas and M. Sloman, "Starfish: Policy driven self-management in wireless sensor networks", in *Proc. Workshop on Software Eng. for Adaptive and Self-Managing Syst.*, Cape Town, South Africa, pp.75–83, 2010.

[35] M. Pajic, A. Chernoguzov and R. Mangharam, "Robust architectures for embedded wireless network control and actuation", *ACM Trans. Embedded Computing Syst.*, vol. 11, no. 4, pp. 82–106, 2012.

[36] C. L. Fok, G. C. Roman and C. Lu, "Agilla: A mobile agent middle-ware for self-adaptive wireless sensor networks", *ACM Trans. Autonomous and Adaptive Syst.*, vol. 4, no. 6, pp.16–42, 2009.

[37] S. B. Meskina, N. Doggaz, M. Khalgui and Z. Li, "Multiagent framework for smart grids recovery", *IEEE Trans. Syst., Man, Cybern., Syst.*, doi: 10.1109/TSMC.2016.2573824.

[38] H. Grichi, O. Mosbahi, M. Khalgui, and Z. Li, "RWiN: New methodology for the development of reconfigurable WSN", *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 109–125, 2017.

[39] H. Grichi, O. Mosbahi, M. Khalgui, and Z. Li, "New power-oriented methodology for dynamic resizing and mobility of reconfigurable wireless sensor networks", *IEEE Trans. Syst., Man, Cybern., Syst.*, 2017.

[40] M. O. Ben Salem, O. Mosbahi, M. Khalgui, Z. Jlalia, G. Frey and M. Smida, "BROMETH: Methodology to design safe reconfigurable medical robotic systems", *Int. J. of Medical Robotics and Comput. Assisted Surgery*, DOI: 10.1002/rcs.1786.

[41] O. Karoui, M. Khalgui, A. Koubaa, E. Guerfala, Z. Li and E. Tovar, "Dual mode for vehicular platoon safety: Simulation and formal verification", *Inform. Sci.*, vol. 402, pp. 216–232, 2017.

[42] X. Wang, I. Khemaissia, M. Khalgui, Z.W. Li, O. Mosbahi and M.C. Zhou, "Dynamic low-power reconfiguration of real-time systems with periodic and probabilistic tasks", *IEEE Trans. Automation Sci. and Eng.* vol. 12, no. 1, pp. 258–271, 2015.

[43] R. Idriss, A. Loukil, M. Khalgui, Z. Li and A. Al-Ahmari, "Filtering and intrusion detection approach for secured reconfigurable mobile systems", *Journal of Electrical Engineering and Technology* vol. 12, no. 5, pp. 2051–2066, 2017.

[44] W. Housseyni, O. Mosbahi, M. Khalgui, Z. Li, and L. Yin, "Multi-agent architecture for distributed adaptive scheduling of reconfigurable real-time tasks with energy harvesting constraints", *IEEE Access journal* vol. 6, DOI: 10.1109/ACCESS.2017.2781459, pp. 2068–2084, 2017.

[45] I. Ghribi, R. Ben Abdallah, M. Khalgui, Z. Li, K. Alnowibet and M. Platzner, "R-codesign: codesign methodology for real-time reconfigurable embedded systems under energy constraints", *IEEE Access journal* vol. 6, DOI: 10.1109/ACCESS.2018.2799852, pp. 14078–14092, 2018.

[46] W. Lakhdhar, R. Mzid, M. Khalgui, Z. Li, G. Frey and A. Al-Ahmari, "Multi-objective optimization approach for a portable development of reconfigurable real-time systems: from specification to implementation", *IEEE Trans. Syst., Man, Cybern., Syst.* vol. PP, no. 99, DOI: 10.1109/TSMC.2017.2781460, pp. 1–15, 2018.

[47] M. G. Abidi, M. Ben Smida, M. Khalgui, Z. Li and N. Wu, "Multi-agent oriented solution for forecasting-based control strategy with load priority of microgrids in an island mode case study: Tunisian petroleum platform", *Electric Power Systems Research* vol. 152, DOI: https://doi.org/10.1016/j.epsr.2017.07.013, pp. 411–423, 2017.

[48] O. Karoui, E. Guerfala, A. Koubaa, M. Khalgui, E. Tovard, N. Wu, A. Al-Ahmari and Z. Li, "Performance evaluation of vehicular platoons using webots", *IET Intelligent Transport Systems* vol. 11, no. 8, DOI: 10.1049/iet-its.2017.0036, pp. 441–449, 2017.

[49] Y. F. Chen, Z. W. Li, K. Barkaoui, N. Q. Wu and M. C. Zhou, "Compact supervisory control of discrete event systems by Petri nets with data inhibitor arcs", *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 364–379, 2017.

[50] X. Wang, Z. W. Li and W.M. Wonham, "Dynamic multiple-period reconfiguration of real-time scheduling based on timed DES supervisory control", *IEEE Trans. Ind. Informatics*, vol. 12, no. 1, pp. 101–111, 2016.

[51] M. Uzam, Z. W. Li, G. Gelen, and R. S. Zakariyya, "A divide-and-conquer-method for the synthesis of liveness enforcing supervisors for flexible manufacturing systems", *J. of Intelligent Manufacturing*, vol. 27, no. 5, pp. 1111–1129, 2016.